

Amendments to the Claims

Please amend claims 1, 10, 12, 13, 14, 17, 19, 20, 29, 32, 33, 36 and 38, as shown below.

Please cancel claims 3, 4, 9, 16, 18, 22, 23, 28, 35 and 37. The currently pending claims after amendment are listed below.

1 1. (Currently Amended) A method for increasing communication efficiency in a multi-
2 processor system, comprising:
3 snooping, at a processor having a transition cache and at least one level of cache associated
4 therewith, a first command on a system bus, said system bus providing communication between
5 processors in said multi-processor system, said first command requesting invalidation of a cache
6 line;
7 generating a second command in response to said first command at one of said levels of
8 cache which stores said cache line if a memory image coherency state for said cache line indicates
9 that said cache line includes modified data, said second command instructing that said cache line
10 be castback;
11 transferring said second command and said cache line from said one of said levels of cache
12 to said transition cache in response to said first command;
13 invalidating said cache line in each level of cache associated with said processor that stores
14 said cache line;
15 snooping a system response to said first command at said processor; and
16 processing said second command at said processor based on said system response to said
17 first command;
18 wherein said processing step converts said second command to a third command in said
19 transition cache if said system response to said first command is a retry, said third command
20 requesting that said cache line be stored in a main memory of said multi-processor system.

1 2. (Original) The method of claim 1, wherein said processing step discards said second
2 command and said cache line from said transition cache when said system response to said first
3 command is not a retry.

3-9 (Cancelled)

1 10. (Currently Amended) ~~The A method of claim 9~~ for increasing communication efficiency
2 in a multi-processor system, comprising:

3 storing a non-exclusive command associated with a real address in a transition cache of a
4 processor;

5 snooping, at said processor, a command on a system bus providing communication
6 between processors in said multi-processor system, said snooped command being associated with
7 said real address;

8 determining, at said transition cache, whether data has started to arrive at said transition
9 cache in response to said non-exclusive command; and

10 generating a snoop response at said transition cache to said snooped command based on a
11 result of said determining step;

12 wherein said generating step does not generate a retry snoop response when said
13 determining step determines that data has not started to arrive at said transition cache in response
14 to said non-exclusive command.

1 11. (Original) The method of claim 10, wherein said generating step generates a retry snoop
2 response when said determining step determines that data has started to arrive at said transition
3 cache in response to said non-exclusive command.

12. (Currently Amended) The method of claim 10, further comprising:
updating a memory coherency image state for said non-exclusive command at said
transition cache based on said snoop command when said determining step determines that data
has not started to arrive at said transition cache in response to said non-exclusive command.

13. (Currently Amended) ~~The A method of claim 9~~ for increasing communication efficiency
in a multi-processor system, comprising:
storing a non-exclusive command associated with a real address in a transition cache of a
processor;
snooping, at said processor, a command on a system bus providing communication
between processors in said multi-processor system, said snoop command being associated with
said real address;
determining, at said transition cache, whether data has started to arrive at said transition
cache in response to said non-exclusive command; and
generating a snoop response at said transition cache to said snoop command based on a
result of said determining step;
wherein said generating step generates a retry snoop response when said determining step
determines that data has started to arrive at said transition cache in response to said non-exclusive
command.

1 14. (Currently Amended) A method for increasing communication efficiency in a multi-
2 processor system, comprising:

3 receiving, at a processor, a first command on a system bus, said system bus providing
4 communication between processors in said multi-processor system, said first command requesting
5 a cache line;

6 transferring said requested cache line from a cache associated with said processor to a
7 transition cache in said processor as part of a response to said first command;

8 updating a memory coherency image state associated with said cache line in each cache
9 associated with said processor that stores said cache line;

10 snooping a system response to said first command on said system bus; and

11 processing said requested cache line at said processor based on said system response;

12 wherein said processing step converts said response to said first command into a second
13 command for writing said requested cache line in a main memory of said multi-processor system
14 when said system response to said first command is a retry and said memory coherency image
15 state for said requested cache line in said cache which transferred said requested cache line to said
16 transition cache indicated modified data in said requested cache line prior to said updating step.

1 15. (Original) The method of claim 14, wherein said processing step outputs said requested
2 cache line on said system bus when said system response to said first command is not a retry.

16. (Cancelled)

1 17. (Currently Amended) The method of ~~claim 16~~ claim 15, wherein said processing step
2 discards said response to said first command when said system response to said first command is
3 a retry and said memory coherency image state for said requested cache line in said cache which
4 transferred said requested cache line to said transition cache does not indicate modified data in
5 said requested cache line prior to said updating step.

18. (Cancelled)

1 19. (Currently Amended) The method of ~~claim 18~~ claim 14, wherein said processing step
2 discards said response to said first command when said system response to said first command is
3 a retry and said memory coherency image state for said requested cache line in said cache which
4 transferred said requested cache line to said transition cache does not indicate modified data in
5 said requested cache line prior to said updating step.

1 20. (Currently Amended) A multi-processor system, comprising:
2 at least first and second processors;
3 a system bus providing communication between said first and second processors;
4 a bus arbiter generating system responses to commands on said system bus; and wherein
5 said first processor has at least one level of cache associated therewith, a system bus
6 controller controlling communication between said first processor and said system bus, and a
7 transition cache serving as an interface between each level of cache and said system bus
8 controller;
9 one of said levels of cache associated with said first processor stores a cache line having a
10 memory coherency image state indicating that said cache line includes modified data, and
11 generates a castback command and transfers said castback command and a copy of said cache line
12 to said transition cache when said first processor snoops a first command on said system bus that
13 requests invalidation of said cache line; and
14 each level of cache associated with said first processor that stores said cache line
15 invalidates said cache line prior to said first processor snooping a system response to said first
16 command;
17 wherein said transition cache converts said castback command to a second command if
18 said system response to said first command is a retry, said second command requesting that said
19 cache line be stored in a main memory of said multi-processor system.

1 21. (Original) The system of claim 20, wherein said transition cache discards said castback
2 command and said cache line when said system response to said first command is not a retry.

22-28 (Cancelled)

1 29. (Currently Amended) ~~The~~ A multi-processor system of claim 28, comprising:
2 at least first and second processors;
3 a system bus providing communication between said first and second processors;
4 a bus arbiter generating system responses to commands on said system bus; and wherein
5 said first processor includes at least one level of cache associated therewith, a system bus
6 controller controlling communication between said first processor and said system bus, and a
7 transition cache controlling and tracking communication between each level of cache and said
8 system bus controller; and
9 said transition cache determines whether data has started to arrive at said transition cache
10 in response to a non-exclusive command, which is associated with a real address, stored therein
11 when said first processor snoops a command on said system bus which is associated with said real
12 address, and generates a snoop response to said snooped command based on said determination;
13 wherein said transition cache does not generate a retry snoop response when data has not
14 started to arrive at said transition cache in response to said non-exclusive command.

1 30. (Original) The system of claim 29, wherein said transition cache generates a retry snoop
2 response when data has started to arrive at said transition cache in response to said non-exclusive
3 command.

1 31. (Original) The system of claim 29, wherein said transition cache updates a memory
2 coherency image state for said non-exclusive command based on said snooped command when
3 data has not started to arrive at said transition cache in response to said non-exclusive command.

1 32. (Currently Amended) ~~The~~ A multi-processor system of claim 28, comprising:
2 at least first and second processors;
3 a system bus providing communication between said first and second processors;
4 a bus arbiter generating system responses to commands on said system bus; and wherein
5 said first processor includes at least one level of cache associated therewith, a system bus
6 controller controlling communication between said first processor and said system bus, and a
7 transition cache controlling and tracking communication between each level of cache and said
8 system bus controller; and
9 said transition cache determines whether data has started to arrive at said transition cache
10 in response to a non-exclusive command, which is associated with a real address, stored therein
11 when said first processor snoops a command on said system bus which is associated with said real
12 address, and generates a snoop response to said snooped command based on said determination;
13 wherein said transition cache generates a retry snoop response when data has started to
14 arrive at said transition cache in response to said non-exclusive command.

1 33. (Currently Amended) A multi-processor system, comprising:
2 at least first and second processors;
3 a system bus providing communication between said first and second processors;
4 a bus arbiter generating system responses to commands on said system bus; and wherein
5 said first processor has at least one cache associated therewith, a system bus controller
6 controlling communication between said first processor and said system bus, and a transition
7 cache controlling and tracking communication between each cache and said system bus
8 controller;
9 said first processor receives a first command on said system bus requesting a cache line;
10 one of said caches associated with said first processor that stores said requested cache line
11 copies said requested cache line to said transition cache as part of a response to said first
12 command;
13 each cache associated with said first processor, that stores said requested cache line,
14 updates a memory coherency image state associated with said requested cache line prior to
15 snooping a system response to said first command; and
16 said first processor snoops said system response on said system bus to said first command,
17 and processes said requested cache line based on said system response;
18 wherein said first processor converts said response to said first command into a second
19 command for writing said requested cache line in a main memory of said multi-processor system
20 when said system response to said first command is a retry and said memory coherency image
21 state for said requested cache line in said cache which transferred said requested cache line to said
22 transition cache indicated modified data in said requested cache line prior to said updating step.

1 34. (Original) The system of claim 33, wherein said first processor outputs said requested
2 cache line on said system bus when said system response to said first command is not a retry.

35. (Cancelled)

1 36. (Currently Amended) The system of ~~claim 35~~ claim 34, wherein said first processor
2 discards said response to said first command when said system response to said first command is
3 a retry and said memory coherency image state for said requested cache line in said cache which
4 transferred said requested cache line to said transition cache does not indicate modified data in
5 said requested cache line prior to said updating step.

37. (Cancelled)

1 38. (Currently Amended) The system of ~~claim 37~~ claim 33, wherein said first processor
2 discards said response to said first command when said system response to said first command is
3 a retry and said memory coherency image state for said requested cache line in said cache which
4 transferred said requested cache line to said transition cache does not indicate modified data in
5 said requested cache line prior to said updating step.